

Empirical Interpolation of Nonlinear Parametrized Evolution Operators

Outline

Reduced Basis Scheme for Parametrized Evolution Equations

Empirical (Operator) Interpolation

Fréchet derivative of empirical interpolation

Efficient reduced basis generation

- X-greedy
- El-greedy
- POD-greedy
- PODEI-greedy

A posteriori error estimate

Numerical examples

Outlook

Problem: Parametrized Evolution Equation

Analytical Formulation

For $\mu \in \mathcal{P} \subset \mathbb{R}^p$, find $u : [0, T_{\max}] \rightarrow \mathcal{W} \subset L^2(\Omega)$, s.t.

$$u(0) = u_0(\mu), \quad \partial_t u(t) - \mathcal{L}(\mu)[u(t)] = 0$$

plus (parameter dependent) boundary conditions.

Problem: Parametrized Evolution Equation

Analytical Formulation

For $\mu \in \mathcal{P} \subset \mathbb{R}^p$, find $u : [0, T_{\max}] \rightarrow \mathcal{W} \subset L^2(\Omega)$, s.t.

$$u(0) = u_0(\mu), \quad \partial_t u(t) - \mathcal{L}(\mu)[u(t)] = 0$$

plus (parameter dependent) boundary conditions.

Discretization (implicit/explicit with Newton scheme)

For $\mu \in \mathcal{P}$ find $\{u_h\}_{k=0}^K \subset \mathcal{W}_h \subset \mathcal{W}$, s.t.

$$u_h^0 := \mathcal{P}_h[u_0(\mu)], \quad u_h^{k+1} := u_h^{k+1, \nu_{\max}(k)}$$

with Newton iteration

$$u_h^{k+1,0} := u_h^k, \quad u_h^{k+1,\nu+1} := u_h^{k+1,\nu} + \delta_h^{k+1,\nu+1},$$

$$\left(\text{Id} + \Delta t \mathbf{D} \mathcal{L}_{h,I} \big|_{u_h^{k+1,\nu}} \right) \left[\delta_h^{k+1,\nu+1} \right] = u_h^k - u_h^{k+1,\nu} - \Delta t \left(\mathcal{L}_{h,I} \left[u_h^{k+1,\nu} \right] + \mathcal{L}_{h,E} \left[u_h^k \right] \right).$$

Example: FV scheme for Burgers-Equation

Burgers Equation

$$\partial_t u - \nabla \mathbf{v} u^{\mu_1} = 0 \quad (1)$$

with (implicit) finite volume discretization with Engquist Osher flux.

- ▶ Parameter vector $\mu := (\mu_1) \in [1, 2]$.
- ▶ $\Omega = [0, 2] \times [0, 1]$ with purely cyclical boundary conditions
- ▶ end time $T = 0.3$
- ▶ smooth initial data: $u_0(x) = \frac{1}{2}(1 + \sin(2\pi x_1) \sin(2\pi x_2))$
- ▶ rectangular 120×60 grid with $K = 100$ time steps.

Example: FV scheme for Burgers-Equation

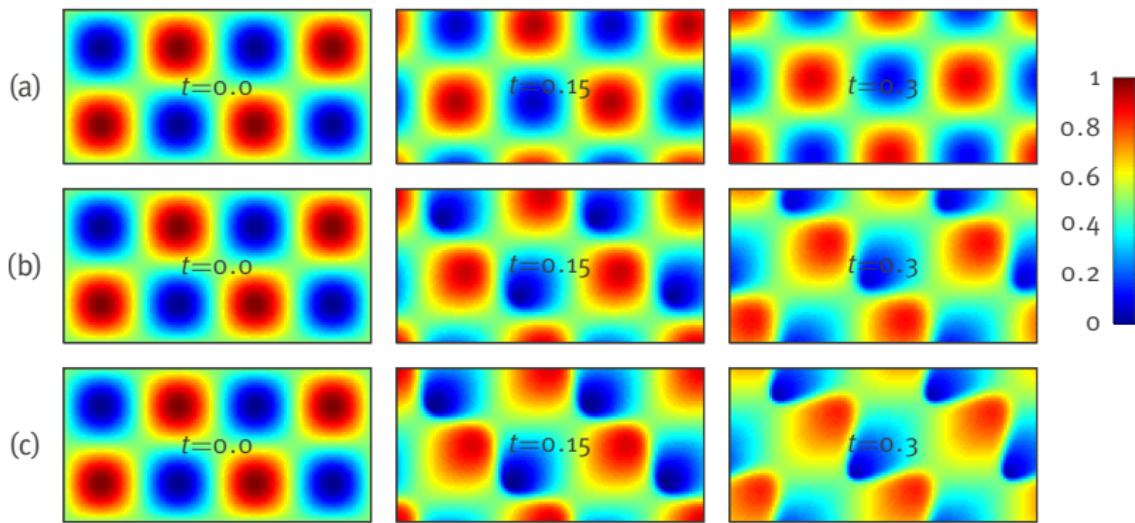
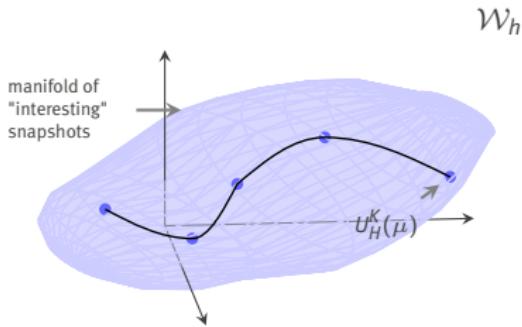


Figure: Illustration of transport for smooth data. Snapshots at different time instants for a) $\mu = 1$ b) $\mu = 1.5$ and c) $\mu = 2$.

Motivation: Reduced Basis Method

RB Scenario:

- ▶ Applications relying on **time-critical** or many **repeated** simulations



Goals:

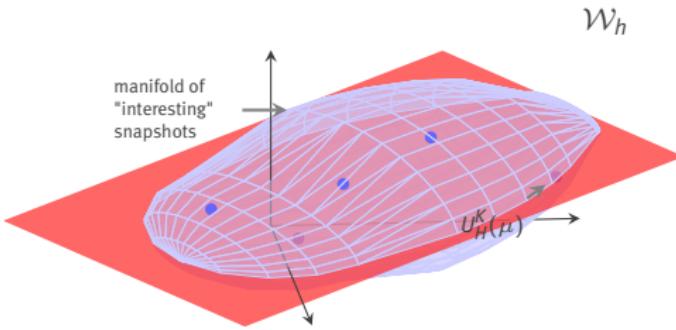
- ▶ **Offline-/Online decomposition**
- ▶ Efficient reduced simulations
- ▶ A posteriori error control

References: [Patera&Rozza, 2006], [Haasdonk et al., 2008]

Motivation: Reduced Basis Method

RB Scenario:

- ▶ Applications relying on **time-critical** or many **repeated** simulations



Goals:

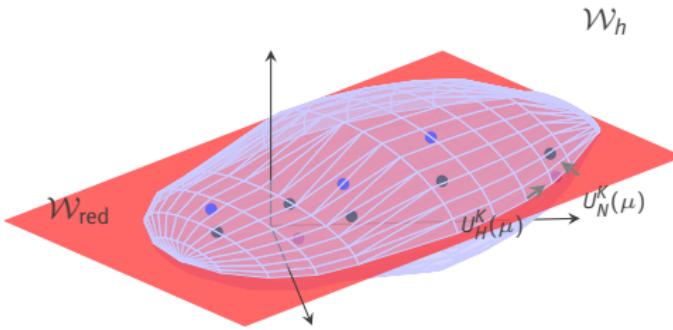
- ▶ **Offline-/Online** decomposition
- ▶ Efficient reduced simulations
- ▶ A posteriori error control

References: [Patera&Rozza, 2006], [Hausdorff et al., 2008]

Motivation: Reduced Basis Method

RB Scenario:

- ▶ Applications relying on **time-critical** or many **repeated** simulations



Goals:

- ▶ Offline-/Online decomposition
- ▶ Efficient **reduced** simulations
- ▶ A posteriori error control

References: [Patera&Rozza, 2006], [Haasdonk et al., 2008]

Reduced basis scheme

Reduced simulation

For $\mu \in \mathcal{P}$ find $\{u_{\text{red}}^k\}_{k=0}^K \subset \mathcal{W}_{\text{red}}$, s.t.

$$u_{\text{red}}^{k+1} := u_{\text{red}}^{k+1, \nu_{\max}(k)}, \quad u_{\text{red}}^0 := \mathcal{P}_{\text{red}}[u_{h,0}(\mu)]$$

with Newton iteration

$$\begin{aligned} u_{\text{red}}^{k+1,0} &:= u_{\text{red}}^k, & u_{\text{red}}^{k+1,\nu+1} &:= u_{\text{red}}^{k+1,\nu} + \delta_{\text{red}}^{k+1,\nu+1}, \\ \left(\text{Id} + \Delta t \mathbf{D}\mathcal{L}_{\text{red},I} |_{u_{\text{red}}^{k+1,\nu}} \right) [\delta_{\text{red}}^{k+1,\nu+1}] &= u_{\text{red}}^k - u_{\text{red}}^{k+1,\nu} - \Delta t \left(\mathcal{L}_{\text{red},I} [u_{\text{red}}^{k+1,\nu}] + \mathcal{L}_{\text{red},E} [u_{\text{red}}^k] \right) \end{aligned}$$

Prerequisites

- ▶ $\mathcal{P}_{\text{red}} : \mathcal{W}_h \rightarrow \mathcal{W}_{\text{red}}$: Galerkin projection onto $\mathcal{W}_{\text{red}} \subset \mathcal{W}_h$
- ▶ $\mathcal{L}_{\text{red},I} := \mathcal{P}_{\text{red}} \circ \mathcal{I}_M \circ \mathcal{L}_{h,I}$: implicit operator evaluation
- ▶ $\mathcal{L}_{\text{red},E} := \mathcal{P}_{\text{red}} \circ \mathcal{I}_M \circ \mathcal{L}_{h,E}$: explicit operator evaluation

Prerequisites

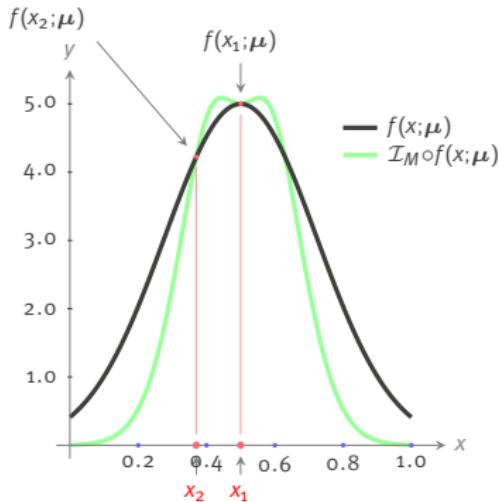
- $\mathcal{P}_{\text{red}} : \mathcal{W}_h \rightarrow \mathcal{W}_{\text{red}}$: Galerkin projection onto $\mathcal{W}_{\text{red}} \subset \mathcal{W}_h$

Empirical interpolation of discrete operators

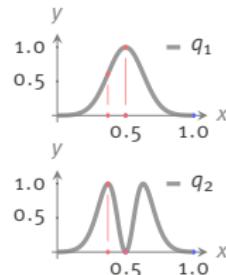
- $\mathcal{L}_{\text{red},I} := \mathcal{P}_{\text{red}} \circ \mathcal{I}_M \circ \mathcal{L}_{h,I}$: implicit operator evaluation
- $\mathcal{L}_{\text{red},E} := \mathcal{P}_{\text{red}} \circ \mathcal{I}_M \circ \mathcal{L}_{h,E}$: explicit operator evaluation

Empirical interpolation [Barrault et al, 2004]

Empirical interpolation for parametrized functions $f(\mu) : \mathbb{R} \rightarrow \mathbb{R}$.



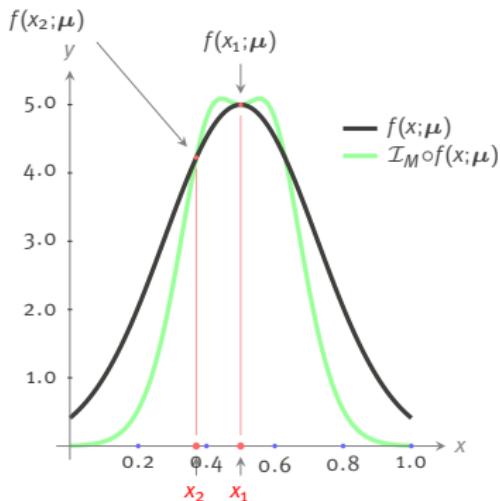
Base functions:



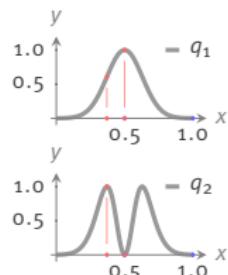
- ▶ “magic points” $\{x_m\}_{m=1}^M$
- ▶ basis functions $\{q_m\}_{m=1}^M$

Empirical interpolation [Barrault et al, 2004]

Empirical interpolation for parametrized discrete operators $\mathcal{L}_h \in \mathcal{W}_h$.



Base functions:

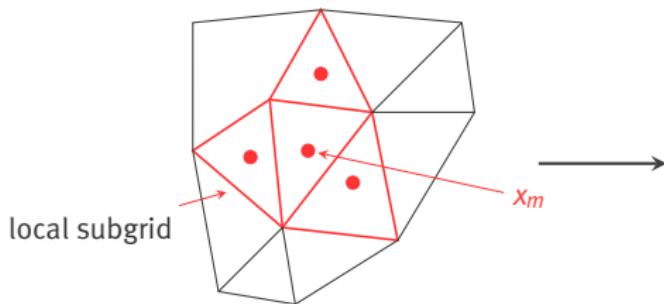


- ▶ “magic points” $\{x_m\}_{m=1}^M$
- ▶ basis functions $\{q_m\}_{m=1}^M$

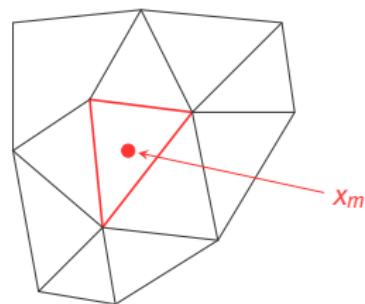
Discrete operators need to have “*H-independent Dof dependence*”.

Empirical interpolation: Subgrids

Restrict $u_h(\mu)$ to subgrid



Evaluate $\mathcal{L}_h(\mu)[u_h(\mu)](x_m)$



Efficient evaluations

The operator evaluations in interpolation points $\mathcal{L}_h(\mu)[\cdot](x_m)$ can be computed efficiently during **online** phase, if

- ▶ the operator has a localized structure (**small stencil**) and
- ▶ the local geometry information is precomputed during **offline** phase.

Empirical interpolation: Details

General operator approximation

Interpolate operator evaluations at “magic points”:

parameter de-
pendent

parameter inde-
pendent

$$\mathcal{L}_h(\mu) \left[u_h^k(\mu) \right] (x_m) = (\mathcal{I}_M \circ \mathcal{L}_h(\mu)) \left[u_h^k(\mu) \right] (x_m) = \sum_{m=1}^M \sigma_m(\mu) q_m(x_m).$$

Empirical interpolation

- ▶ Basis functions q_m are directly computed from operator evaluations
- ▶ for selected parameters μ_m and time steps k_m .
- ▶ They span a collateral reduced basis space $\mathcal{W}_M \subset \mathcal{W}_h$.

Empirical interpolation: Fréchet derivative

Define $l_m(\mu) : \mathcal{W}_h \rightarrow \mathbb{R}$, $u_h \mapsto \mathcal{L}_h(\mu)[u_h](x_m)$

Observation

$$(\mathbf{D} (\mathcal{I}_M [\mathcal{L}_h(\mu)])|_{u_h} [v_h])(x_m) = (\mathbf{D} l_m(\mu)|_{u_h} [v_h])$$

Empirical interpolation: Fréchet derivative

Define $l_m(\mu) : \mathcal{W}_h \rightarrow \mathbb{R}$, $u_h \mapsto \mathcal{L}_h(\mu)[u_h](x_m)$

$$(\mathbf{D}(\mathcal{I}_M[\mathcal{L}_h(\mu)])|_{u_h}[v_h])(x_m) = (\mathbf{D}l_m(\mu)|_{u_h}[v_h])$$

Coefficient functionals

\mathcal{W}_h is finite dimensional

$$\mathbf{D}l_m(\mu)|_{u_h}[v_h] = \sum_{m=1}^M \sum_{i=1}^H \frac{\partial}{\partial \psi_i} l_m(\mu)[u_h](v_{h,i})$$

Empirical interpolation: Fréchet derivative

Define $l_m(\mu) : \mathcal{W}_h \rightarrow \mathbb{R}$, $u_h \mapsto \mathcal{L}_h(\mu)[u_h](x_m)$

$$(\mathbf{D}(\mathcal{I}_M[\mathcal{L}_h(\mu)])|_{u_h}[v_h])(x_m) = (\mathbf{D}l_m(\mu)|_{u_h}[v_h])$$

Coefficient functionals

\mathcal{W}_h is finite dimensional

$$\mathbf{D}l_m(\mu)|_{u_h}[v_h] = \sum_{m=1}^M \sum_{i=1}^H \frac{\partial}{\partial \psi_i} l_m(\mu)[u_h](v_{h,i})$$

$$= \sum_{m=1}^M \sum_{\substack{i \in \mathcal{I}_{x_m}}} \frac{\partial}{\partial \psi_i} l_m(\mu)[u_h](v_{h,i}).$$

\mathcal{L}_h has local stencil

Empirical interpolation: Fréchet derivative

Define $l_m(\mu) : \mathcal{W}_h \rightarrow \mathbb{R}$, $u_h \mapsto \mathcal{L}_h(\mu)[u_h](x_m)$

$$(\mathbf{D}(\mathcal{I}_M[\mathcal{L}_h(\mu)])|_{u_h}[v_h])(x_m) = (\mathbf{D}l_m(\mu)|_{u_h}[v_h])$$

Coefficient functionals

\mathcal{W}_h is finite dimensional

$$\mathbf{D}l_m(\mu)|_{u_h}[v_h] = \sum_{m=1}^M \sum_{i=1}^H \frac{\partial}{\partial \psi_i} l_m(\mu)[u_h](v_{h,i})$$

$$= \sum_{m=1}^M \sum_{\substack{i \in I_{x_m}}} \frac{\partial}{\partial \psi_i} l_m(\mu)[u_h](v_{h,i}).$$

\mathcal{L}_h has local stencil

Note: $\text{card}(I_{x_m}) < C$ for all $m = 1, \dots, M$. \Rightarrow Complexity still $\mathcal{O}(M)$.

Summary: Reduced basis scheme

- ▶ Generate reduced basis space with **POD-Greedy** algorithm:
 $\mathcal{W}_{\text{red}} := \text{span} \{ \varphi_i \}_{i=1}^N \subset \mathcal{W}_h$
- ▶ Reduced model order by Galerkin projection: $\mathcal{P}_{\text{red}} : \mathcal{W}_h \rightarrow \mathcal{W}_{\text{red}}$
- ▶ Offline-/online decomposition of operators:

$$(\mathcal{P}_{\text{red}}[u_{h,0}(\mu)])_n = \underbrace{\sum_{q=1}^{Q_{u0}} \sigma_{u0}^q(\mu)}_{\text{online}} \underbrace{\int_{\Omega} u_0^q \varphi_n}_{\text{offline}} \quad \text{assuming: } u_0(\mu) = \sum_{q=1}^{Q_{u0}} \sigma_{u0}^q(\mu) u_0^q$$

$$\left(\mathcal{L}_{\text{red},I}(\mu) \begin{bmatrix} u_{\text{red}}^k(\mu) \end{bmatrix} \right)_{nm} = \underbrace{\sum_{m=1}^M \left(l_m''(\mu) [u_{\text{red}}] \right)'}_{\text{online}} \underbrace{\int_{\Omega} q_m \varphi_n}_{\text{offline}}$$

$$\left(\mathbf{D}\mathcal{L}_{\text{red},I}(\mu)|_{u_{\text{red}}} [\delta_{\text{red}}] \right)_{nm} = \underbrace{\sum_{m=1}^M \left(\frac{\partial}{\partial \psi_i} l_m'(\mu) [u_{\text{red}}] \right)'}_{\text{online}} \underbrace{\int_{\Omega} q_m \varphi_n}_{\text{offline}}$$

X-greedy algorithm

X-GREEDY(M_{train} , ε_{tol} , Υ_{\max})

– Initialize reduced basis of dimension Υ_0 :

$$\begin{aligned}\mathcal{D}_{\Upsilon_0} &\leftarrow \text{X-INITBASIS0} \\ \Upsilon &\leftarrow \Upsilon_0\end{aligned}$$

repeat

– Find worst approximated parameter:

$$(\mu_{\max}, t_{\max}) \leftarrow \arg \max_{\mu \in M_{\text{train}}} \text{X-ERRORESTIMATE}(\mathcal{D}_{\Upsilon}, \mu, t^k)$$

– Extend reduced basis by snapshot:

$$\begin{aligned}\mathcal{D}_{\Upsilon+1} &\leftarrow \text{X-EXTENDBASIS}(\mathcal{D}_{\Upsilon}, \mu_{\max}, t_{\max}) \\ \Upsilon &\leftarrow \Upsilon + 1\end{aligned}$$

until $\max_{\mu \in M_{\text{train}}} \text{X-ERRORESTIMATE}(\mathcal{D}_{\Upsilon}) \leq \varepsilon_{\text{tol}}$ or $\Upsilon > \Upsilon_{\max}$

return reduced data: \mathcal{D}_{Υ}

- **Extension:** Adaptive extension of Parameter sampling M_{train} .

EI-greedy methods (1/2)

EI-INITBASIS()**return** empty initial basis: $\mathcal{D}_0 \leftarrow \{\}$ **EI-ERRORESTIMATE($(\mathbf{Q}_M, \Sigma_M), \mu, t^k$)**

- Compute exact operator evaluation
 $v_h \leftarrow \mathcal{L}_h[u_h^k(\mu)]$
 - Compute empirical interpolated operator evaluation
 $v_M \leftarrow \mathcal{I}_M \circ \mathcal{L}_h[u_h^k(\mu)]$
- return**
- approximation error:
- $\|v_h - v_M\|.$

EI-greedy methods (1/2)

EI-INITBASIS()

```
return empty initial basis:  $\mathcal{D}_0 \leftarrow \{\}$ 
```

EI-ERRORESTIMATE($(\mathbf{Q}_M, \Sigma_M), \mu, t^k$)

– Compute exact operator evaluation

$$v_h \leftarrow \mathcal{L}_h[u_h^k(\mu)]$$

– Compute empirical interpolated operator evaluation

$$v_M \leftarrow \mathcal{I}_M \circ \mathcal{L}_h[u_h^k(\mu)]$$

return approximation error: $\|v_h - v_M\|.$

Detailed simulation for all trainings parameters needed. **Expensive!**

EI-greedy methods (2/2)

EI-EXTENDBASIS((\mathbf{Q}_M, Σ_M) , μ , t^k)

- Compute exact operator evaluation.
 $v_h \leftarrow \mathcal{L}_h[u_h^k(\mu)]$
- Compute empirical interpolated operator evaluation.
 $v_M \leftarrow \mathcal{I}_M \circ \mathcal{L}_h[u_h^k(\mu)]$
- Compute the residual.

$$r_M \leftarrow v_h - v_M.$$

- Find “magic point” maximizing the residual.

$$x_{M+1} \leftarrow \arg \sup_{x \in \Sigma_h} |r_M(x)|$$

- Normalize to obtain a new basis function.

$$q_{M+1} \leftarrow r_M \cdot \frac{1}{r_M(x_{M+1})}$$

return extended basis data: $\mathcal{D}_{M+1} \leftarrow \left(\{q_m\}_{m=1}^{M+1}, \{x_m\}_{m=1}^{M+1} \right)$

POD-greedy methods

POD-INITBASIS()

return initial reduced basis functions: $\{\varphi_n\}_{n=1}^{N_0}$

POD-ERRORESTIMATE($\{\varphi_n\}_{n=1}^N, \mu, t^k$)

return error estimate: $\eta_{N,M}^k(\mu) \geq \|u_{\text{red}}^k(\mu) - u_h^k(\mu)\|$

POD-EXTENDBASIS($\{\varphi_n\}_{n=1}^N, \mu_{\max}, \cdot$)

- Compute trajectory $\{u_h^k(\mu_{\max})\}_{k=0}^K$.
- Compute new basis function with POD and Galerkin projection \mathcal{P}_{red} projecting onto span $\{\varphi_n\}_{n=1}^N$:

$$\varphi_{N+1} \leftarrow \text{POD} \left(\{u_h^k(\mu_{\max}) - \mathcal{P}_{\text{red}}[u_h^k(\mu_{\max})]\}_{k=0}^K \right)$$

return extended reduced basis: $\{\varphi_n\}_{n=1}^{N+1}$

POD-greedy methods

POD-INITBASIS()

return initial reduced basis functions: $\{\varphi_n\}_{n=1}^{N_0}$

POD-ERRORESTIMATE($\{\varphi_n\}_{n=1}^N, \mu, t^k$)

return *error estimate*: $\eta_{N,M}^k(\mu) \geq \|u_{\text{red}}^k(\mu) - u_h^k(\mu)\|$

POD-EXTENDBASIS($\{\varphi_n\}_{n=1}^N, \mu_{\max}, \cdot$)

- Compute trajectory $\{u_h^k(\mu_{\max})\}_{k=0}^K$.
- Compute new basis function with POD and Galerkin projection \mathcal{P}_{red} projecting onto span $\{\varphi_n\}_{n=1}^N$:

$$\varphi_{N+1} \leftarrow \text{POD} \left(\{u_h^k(\mu_{\max}) - \mathcal{P}_{\text{red}}[u_h^k(\mu_{\max})]\}_{k=0}^K \right)$$

return extended reduced basis: $\{\varphi_n\}_{n=1}^{N+1}$

Remarks: EI-greedy + POD-greedy

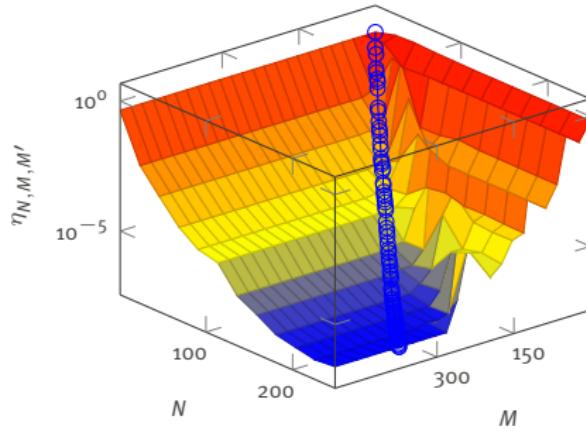
- ▶ “EI-greedy” has to be computed before “POD-greedy” with detailed simulations for all trainings parameters.
- ▶ A priori it is unknown, how to choose the error tolerance ε_{tol} for the “EI-greedy”.
- ▶ Number of EI base functions might be too large.

Remarks: EI-greedy + POD-greedy

- ▶ “EI-greedy” has to be computed before “POD-greedy” with detailed simulations for all trainings parameters.
- ▶ A priori it is unknown, how to choose the error tolerance ε_{tol} for the “EI-greedy”.
- ▶ Number of EI base functions might be too large.

Alternative: “PODEI-greedy”

Motivation: PODEI-greedy





PODEI-greedy methods (1 of 2)

PODEI-INITBASIS()

```
– Generate small empirical interpolation basis:  
     $(\mathbf{Q}_{M_{\text{small}}}, \Sigma_{M_{\text{small}}}) \leftarrow \text{EI-GREEDY}(M_{\text{train}}^{\text{coarse}}, \varepsilon_{\text{tol,small}}, M_{\text{small}})$   
– Compute initial reduced basis:  
     $\{\varphi_n\}_{n=0}^{N_0} \leftarrow \text{RB-INITBASIS}()$   
return initial bases data:  $\mathcal{D}_1 \leftarrow \{\varphi_n\}_{n=1}^{N_0} \cup (\mathbf{Q}_{M_{\text{small}}}, \Sigma_{M_{\text{small}}})$ 
```

PODEI-ERRORESTIMATE(\mathcal{D}_T, μ, t^k)

```
return reduced basis error estimate:  $\eta_{N,M}^k(\mu)$ 
```

PODEI-greedy methods (2 of 2)

PODEI-EXTENDBASIS($\mathcal{D}_\gamma, \mu_{\max}, t^k$)

Reduced data \mathcal{D}_γ comprises $\mathcal{D}_N^{RB} := \{\varphi_n\}_{n=1}^N$ and $\mathcal{D}_M^{EI} := (\mathbf{Q}_M, \Sigma_M)$

- Extend EI basis: $\mathcal{D}_{M+1}^{EI} \leftarrow EI\text{-EXTENDBASIS}(\mathcal{D}_M^{EI}, \mu_{\max}, t^k)$
- Extend RB basis: $\mathcal{D}_{N+1}^{RB} \leftarrow RB\text{-EXTENDBASIS}(\mathcal{D}_N^{RB}, \mu_{\max}, t^k)$
- Discard extended RB if error increases:

if $\eta_{N-1, M-1}^k(\mu_{\max}) \geq \max_{(\mu, t) \in M_{\text{train}}} \eta_{N, M}^k(\mu)$ then

return extended basis data: $\mathcal{D}_{\gamma+1} \leftarrow (\mathcal{D}_N^{RB}, \mathcal{D}_{M+1}^{EI})$

else

return extended basis data: $\mathcal{D}_{\gamma+1} \leftarrow (\mathcal{D}_{N+1}^{RB}, \mathcal{D}_{M+1}^{EI})$

end if

A posteriori error estimator

Two contributions:

- ▶ Projection error on \mathcal{W}_{red}
- ▶ Interpolation errors of operators by empirical interpolation depending on \mathcal{W}_M and T_M

A posteriori error estimator

Theorem (A posteriori error estimator)

Assumptions:

- ▶ Operators and $\mathbf{D}\mathcal{L}_{h,I}$ are Lipschitz-continuous.
- ▶ $\mathbf{D}\mathcal{L}_{h,I}$ has bounded inverse
- ▶ Empirical interpolations exact for larger CRB space $\mathcal{W}_{M+M'}$ and $\mathcal{P}_h[u_0(\mu)] \in \mathcal{W}_{red}$

Then:

$$\|u_{red}^k(\mu) - u_h^k(\mu)\| \leq \eta_{N,M}^k(\mu) \quad \text{with } \eta_{N,M}(\mu) := \sum_{i=0}^{k-1} \bar{\eta}^{i,\nu_{\max}(i)}$$

recursively defined through the Newton step error estimator

$$\begin{aligned} \bar{\eta}^{k+1,\nu+1} := \Delta t & \left(C_1 \bar{\eta}^{k_1,\nu} + C_2 \bar{\eta}^{k+1,\nu} + \|\delta_{red}^{k+1,\nu+1}\| + \right. \\ & \left. \|R_{D,M}^{k+1,\nu+1}\| + \|R_{I,M}^{k+1,\nu}\| + \|R_{E,M}^{k+1,0}\| + \|R^{k+1,\nu}\| \right). \end{aligned}$$

A posteriori error estimator

Theorem (A posteriori error estimator cont.)

Then:

$$\|u_{red}^k(\mu) - u_h^k(\mu)\| \leq \eta_{N,M}^k(\mu) \quad \text{with } \eta_{N,M}(\mu) := \sum_{i=0}^{k-1} \bar{\eta}^{i,\nu_{\max}(i)}$$

recursively defined through the Newton step error estimator

$$\begin{aligned} \bar{\eta}^{k+1,\nu+1} := & \Delta t \left(C_1 \bar{\eta}^{k_1,\nu} + C_2 \bar{\eta}^{k+1,\nu} + \|\delta_{red}^{k+1,\nu+1}\| + \right. \\ & \left. \|R_{D,M}^{k+1,\nu+1}\| + \|R_{I,M}^{k+1,\nu}\| + \|R_{E,M}^{k+1,0}\| + \|R^{k+1,\nu}\| \right). \end{aligned}$$

The residuals $R_{*,M}$ measure the empirical interpolation error, e.g.

$$R_{I,M}^{k+1,\nu} := \sum_{m=M}^{M+M'} l_m' \left[u_{red}^{k+1,\nu} \right] \xi_m$$

A posteriori error estimator

Theorem (A posteriori error estimator cont.)

The residual R^k deals with the projection error onto the reduced basis space and its L^2 -norm can be computed:

$$\begin{aligned}\Delta t^2 \|R^{k+1}\|^2 &= \langle \Delta t R^{k+1}, \Delta t R^{k+1} \rangle \\ &= (\mathbf{a}^{k+1} - \mathbf{a}^k)^T \mathbf{M} (\mathbf{a}^{k+1} - \mathbf{a}^k)^T \\ &\quad + 2\Delta t \left(\mathbf{l}_I [\mathbf{a}^{k+1}] + \mathbf{l}_E [\mathbf{a}^k] \right)^T \mathbf{C} (\mathbf{a}^{k+1} - \mathbf{a}^k) \\ &\quad + \Delta t^2 \left(\mathbf{l}_I [\mathbf{a}^{k+1}] + \mathbf{l}_E [\mathbf{a}^k] \right)^T \mathbf{X} \left(\mathbf{l}_I [\mathbf{a}^{k+1}] + \mathbf{l}_E [\mathbf{a}^k] \right).\end{aligned}$$

Example: I. Burgers Equation

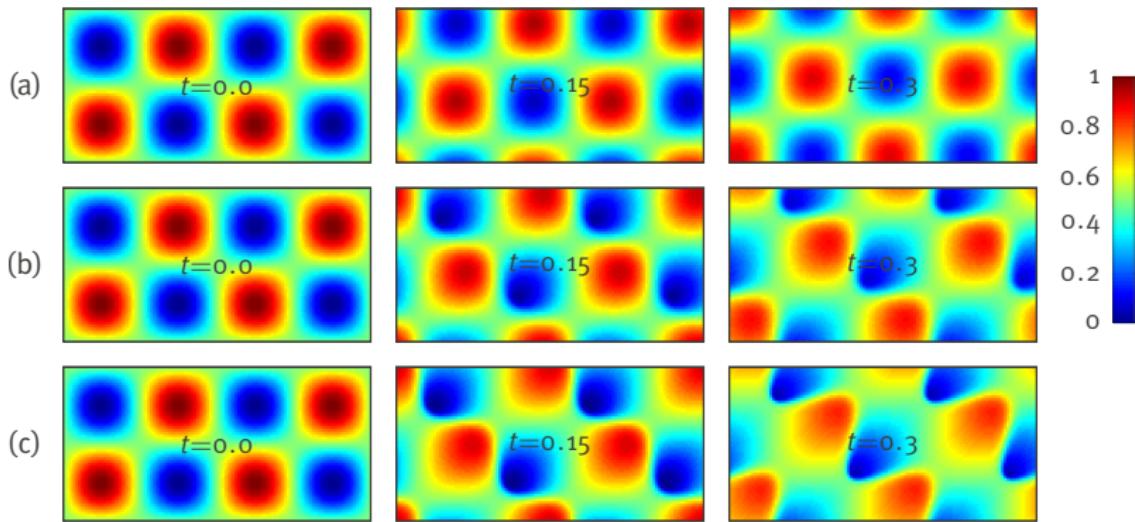
Burgers Equation

$$\partial_t u - \nabla \mathbf{v} u^{\mu_1} = 0 \quad (2)$$

with (implicit) finite volume discretization with Engquist Osher flux.

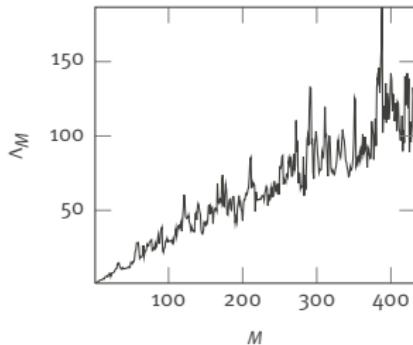
- ▶ Parameter vector $\mu := (\mu_1) \in [0, 2]$.
- ▶ $\Omega = [0, 2] \times [0, 1]$ with purely cyclical boundary conditions
- ▶ end time $T = 0.3$
- ▶ smooth initial data: $u_0(x) = \frac{1}{2}(1 + \sin(2\pi x_1) \sin(2\pi x_2))$
- ▶ rectangular 120×60 grid with $K = 100$ time steps.

Example I: Solution snapshots



Example I: Empirical interpolation of $\mathcal{L}_{h,I}$

(a) Lebesgue constant



(b) El-greedy

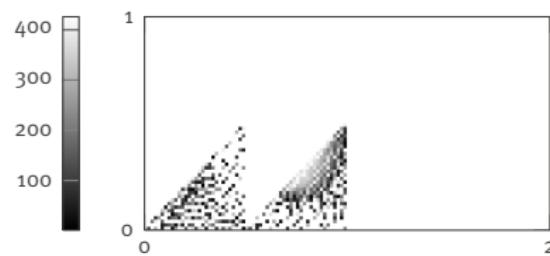


Illustration of interpolation DOF selection for Burgers problem. DOFs corresponding to darker points are selected first.

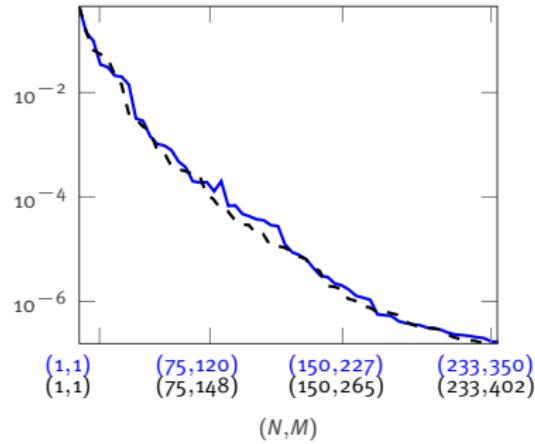
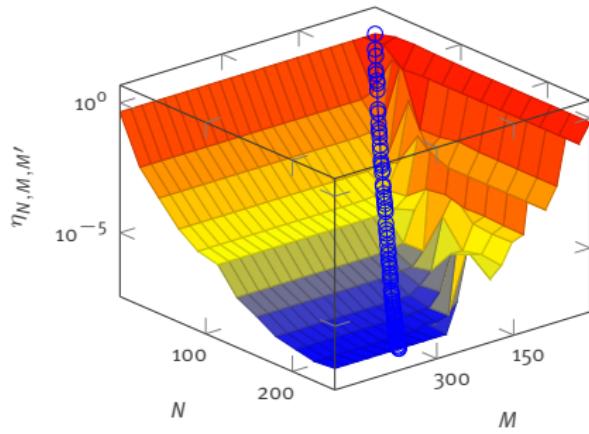
Example I: Table

- ▶ $\dim(\mathcal{W}_h)$ 9600
- ▶ $\nu_{\max} \approx 1 - 20$
- ▶ $\#M_{\text{train}}$ 28

N	M	$\varnothing\text{-runtime}[s]$	max. error	$\varnothing\text{-offline time}[h]$
7,200	0	90.01	0.00	0
42	83	4.42	$1.15 \cdot 10^{-3}$	0.96
83	166	6.23	$6.03 \cdot 10^{-5}$	1.34
125	250	8.99	$7.43 \cdot 10^{-6}$	1.74
166	333	11.6	$8.33 \cdot 10^{-7}$	2.23
208	416	15.64	$2.47 \cdot 10^{-7}$	2.78
249	499	19.56	$2.38 \cdot 10^{-7}$	3.4

N	M	$\varnothing\text{-runtime}[s]$	max. error	$\varnothing\text{-offline time}[h]$
0	-1	90.01	0.00	0
42	72	4.44	$1.73 \cdot 10^{-3}$	0.54
83	144	6.04	$5.74 \cdot 10^{-5}$	1.09
125	216	8.37	$7.30 \cdot 10^{-6}$	1.55
167	288	11.92	$7.63 \cdot 10^{-7}$	2.08
208	360	15.08	$2.31 \cdot 10^{-7}$	2.69
233	402	16.48	$1.55 \cdot 10^{-7}$	3.27

Example I: Error landscape



Example II: Nonlinear Diffusion

Nonlinear Diffusion

Problem definition:

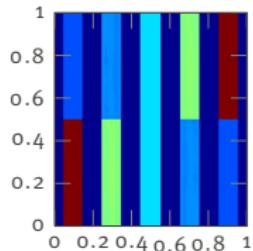
$$\partial_t u - m \Delta u^p = 0 \quad \text{in } \Omega \times [0, 1], \quad u(\cdot, 0) = c_0 + u_0 \quad \text{on } \Omega \times \{0\}$$

with (implicit) finite volume discretization with Engquist–Osher flux

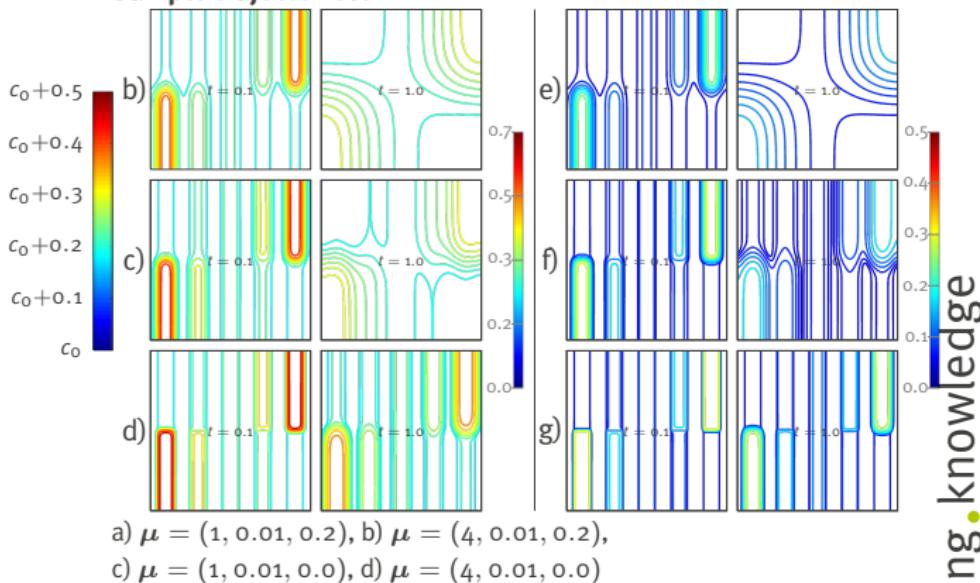
- ▶ $\Omega = [0, 1]^2$ with homogeneous boundary conditions
- ▶ rectangular 100x100 grid with $K = 80$ time steps.
- ▶ Parametrization $\mu = (p, m, c_0) \in [1, 5] \times [0, 0.01] \times [0, 0.2]$

Example II: Solution snapshots

Initial data:



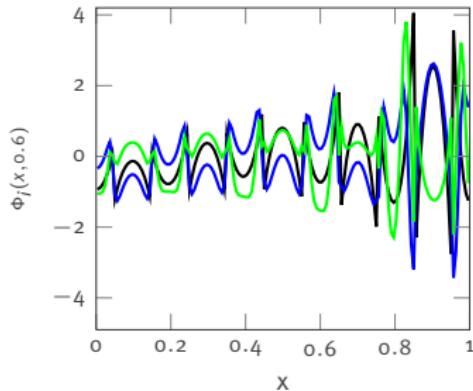
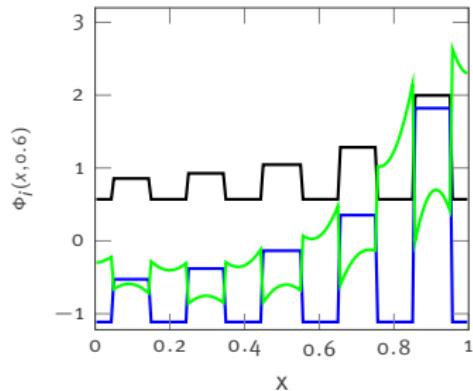
Sample trajectories:



- a) $\mu = (1, 0.01, 0.2)$, b) $\mu = (4, 0.01, 0.2)$,
c) $\mu = (1, 0.01, 0.0)$, d) $\mu = (4, 0.01, 0.0)$

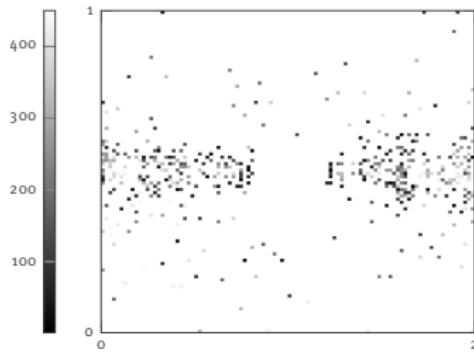
Example II: Reduced basis functions

Slices of first six reduced basis functions at $y = 0.6$

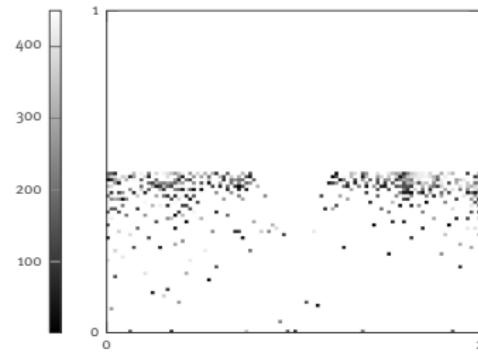


Example II: Empirical interpolation of $\mathcal{L}_{h,I}$

a) PODEI-Greedy



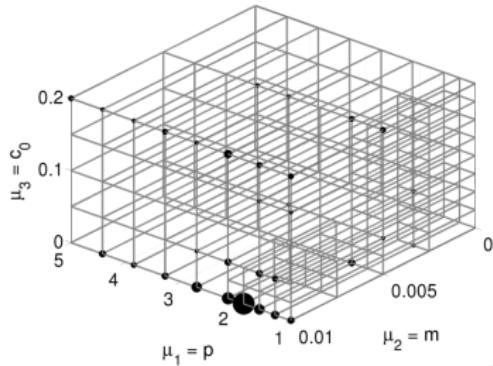
b) EI-Greedy + POD-Greedy



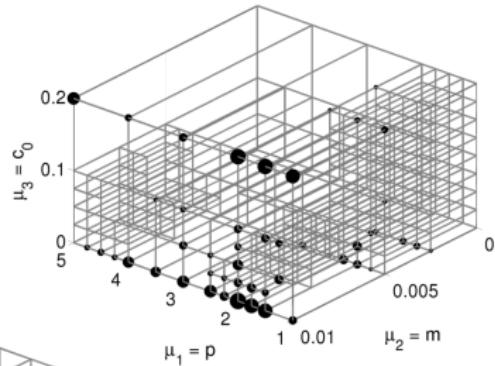
Subgrid at interpolation points

Example II: Empirical interpolation of $\mathcal{L}_{h,I}$

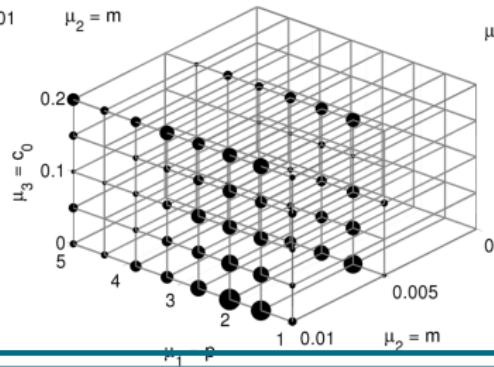
a) Parameter selection POD-Greedy



b) Parameter selection PODEI-Greedy

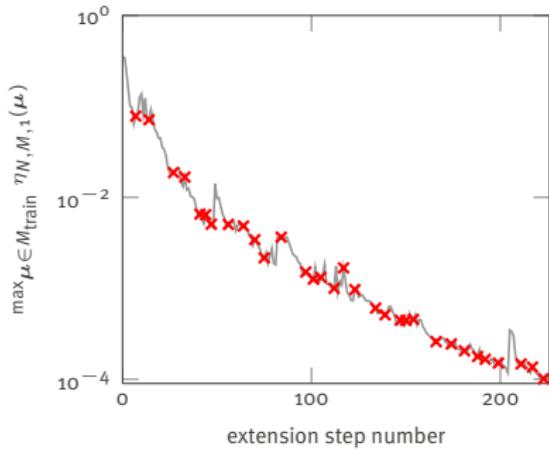


c) Parameter selection El-Greedy

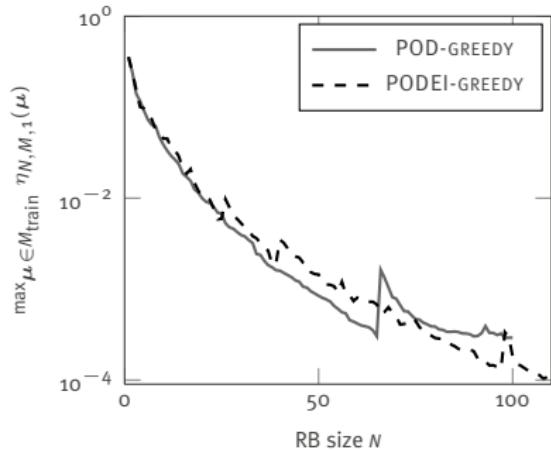


Example II: Greedy error convergence

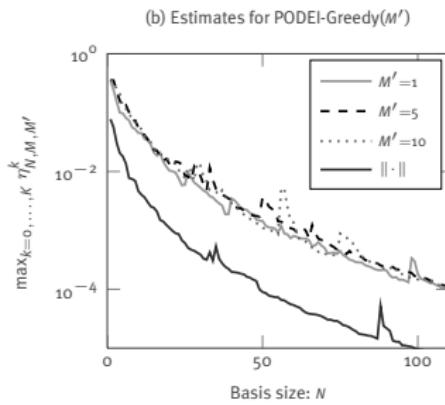
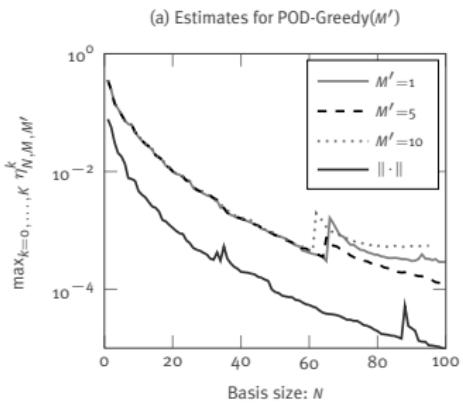
(a) POD-EI-greedy basis discards



(b) X-greedy error decrease



Example II: Greedy error convergence



Example II: Numerical results

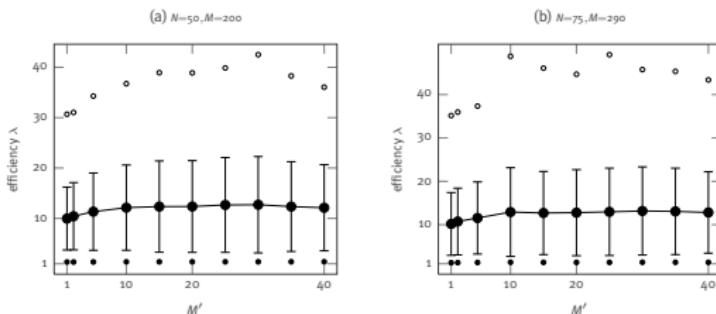
- ▶ $\dim(\mathcal{W}_h) \approx 10000$
- ▶ $\nu_{\max} \approx 1 - 20$
- ▶ $\#M_{\text{train},o} = 27$.
- ▶ $\#M_{\text{train}} = 305$

N	M	$\bar{\theta}$ -runtime[s]	max. error	$\bar{\theta}$ -offline time[h]
0	0	55.38	0.00	0
17	71	1.57	$3.56 \cdot 10^{-3}$	1.35
33	142	1.95	$8.33 \cdot 10^{-4}$	1.67
50	213	2.51	$2.08 \cdot 10^{-4}$	2.07
66	283	3.19	$5.88 \cdot 10^{-5}$	2.43
83	354	4.07	$5.55 \cdot 10^{-5}$	2.88
99	425	5.3	$4.06 \cdot 10^{-5}$	3.3

N	M	$\bar{\theta}$ -runtime[s]	max. error	$\bar{\theta}$ -offline time[h]
0	-1	55.38	0.00	0
19	72	1.61	$3.01 \cdot 10^{-3}$	0.16
37	143	2.07	$7.90 \cdot 10^{-4}$	0.45
56	215	2.67	$1.66 \cdot 10^{-4}$	1.01
74	286	3.6	$6.36 \cdot 10^{-5}$	1.69
93	358	4.83	$3.54 \cdot 10^{-5}$	2.72
111	429	6.55	$1.96 \cdot 10^{-5}$	4.02

Example II: A posteriori error estimator

Efficiency of error estimator η : $\lambda(\mu) := \frac{\eta(\mu)}{\|u_h(\mu) - u_{\text{red}}(\mu)\|}$



Error bar plot showing mean and standard deviation of error estimator efficiency over a sample of 100 random parameters for different values of M' . The dots indicate the minimum (●) and maximum (○) efficiency.



Numerical results

- ▶ Problems are implemented with our software package RBmatlab (<http://www.morepas.org/software>).
- ▶ Computations are executed on compute nodes of the PALMA cluster at the university of Münster with Intel Xeon Westmere X5650, 2,67 GHz processors and 24 GB RAM per node.

Outlook

Conclusion

- ▶ Model order reduction of general (scalar) parametrized evolution schemes
- ▶ with reduced basis methods and empirical interpolation for discrete operators
- ▶ Rigorous error control via a posteriori error estimator is possible.

Outlook

Conclusion

- ▶ Model order reduction of general (scalar) parametrized evolution schemes
- ▶ with reduced basis methods and empirical interpolation for discrete operators
- ▶ Rigorous error control via a posteriori error estimator is possible.

Future work

- ▶ Dealing with steep gradients in solution snapshots (non-linear reduced bases?)
- ▶ Variable time step width
- ▶ 2-Phase flow system
- ▶ Improve software

References

-  [Patera&Rozza, 2006] A. Patera and G. Rozza.
Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations.
Version 1.0, Copyright MIT 2006, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering
-  [Haasdonk&Ohlberger, 2008] B. Haasdonk und M. Ohlberger
Reduced Basis Method for Finite Volume Approximations of Parametrized Linear Evolution Equations
M2AN, Math. Model. Numer. Anal. 42, 277–302, 2008
-  [Haasdonk et al., 2008] B. Haasdonk, M. Ohlberger und G. Rozza.
A Reduced Basis Method for Evolution Schemes with Parameter-Dependent Explicit Operators
ETNA, Electronic Transactions on Numerical Analysis, vol. 32, 2008
-  [Barrault et al., 2004] M. Barrault, M., Y. Maday, N. Nguyen and A. Patera,
An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations
C. R. Math. Acad. Sci. Paris Series I, 2004, 339, 667-672
-  [Drohmann et al., 2009] M. Drohmann, B. Haasdonk and M. Ohlberger,
Reduced Basis Method for Finite Volume Approximation of Evolution Equations on Parametrized Geometries
Proceedings of ALGORITMY 2009, 111–120
-  [Drohmann et al., 2010] M. Drohmann, B. Haasdonk and M. Ohlberger,
Reduced Basis Approximation for Nonlinear Parametrized Evolution Equations based on Empirical Operator Interpolation
FB 10, Universität Münster num. 10/02 - N Preprint - october 2010

Software concepts (DUNE-RB/RBMATLAB interface)

Goals:

- ▶ Use C++ software DUNE-RB for high-dimensional computations and RBMATLAB for reduced basis algorithms
- ▶ Access to open source implementations of “real world” problems. (Several available in DUNE)
- ▶ Testing the reduced basis methods on these implementations.

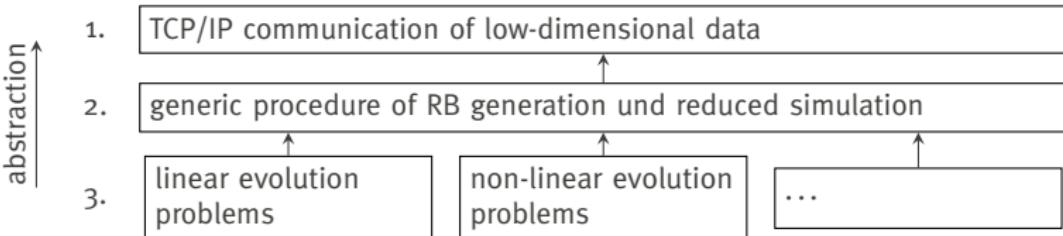
For further information see: <http://morepas.org>

Software concepts (DUNE-RB/RBMATLAB interface)

Goals:

- ▶ Use C++ software DUNE-RB for high-dimensional computations and RBMATLAB for reduced basis algorithms
- ▶ Access to open source implementations of “real world” problems. (Several available in DUNE)
- ▶ Testing the reduced basis methods on these implementations.

Illustration of interface concept:



For further information see: <http://morepas.org>

Software concepts (DUNE-RB/RBMATLAB interface)

Goals:

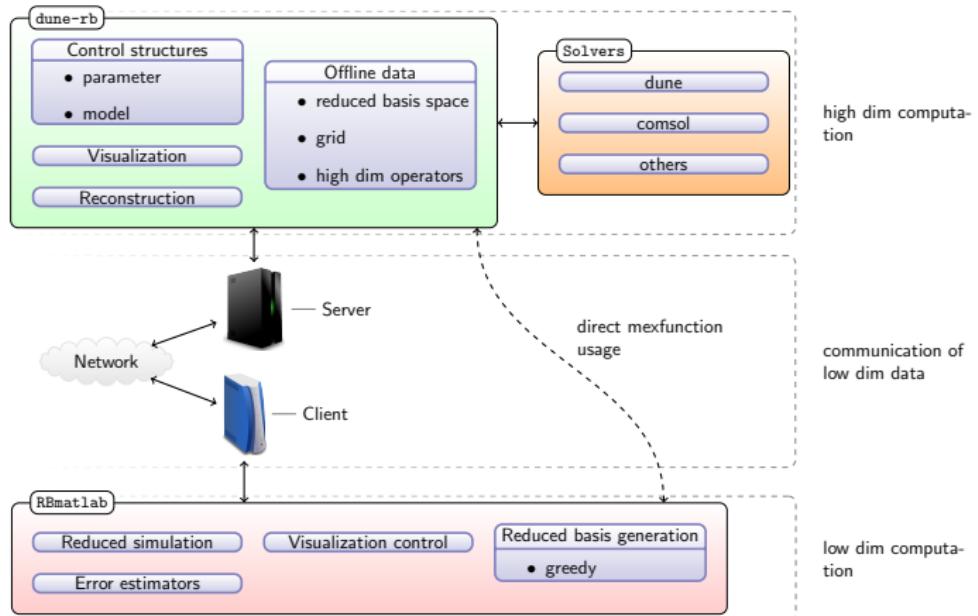
- ▶ Use C++ software DUNE-RB for high-dimensional computations and RBMATLAB for reduced basis algorithms
- ▶ Access to open source implementations of “real world” problems. (Several available in DUNE)
- ▶ Testing the reduced basis methods on these implementations.

Status:

- ▶ Communication interface between RBMATLAB and DUNE-RB exists
- ▶ Example implementation: linear heat equation in Dune affinely parameter dependent structure. (No empirical interpolation necessary)
- ▶ Empirical interpolation of simple operators.

For further information see: <http://morepas.org>

Software concepts (DUNE-RB/RBMATLAB interface)



For further information see: <http://morepas.org>